

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-238030

(43)Date of publication of application : 31.08.1999

(51)Int.Cl. G06F 13/14
G06F 13/36

(21)Application number : 10-038841

(71)Applicant : MITSUBISHI ELECTRIC CORP

(22)Date of filing : 20.02.1998

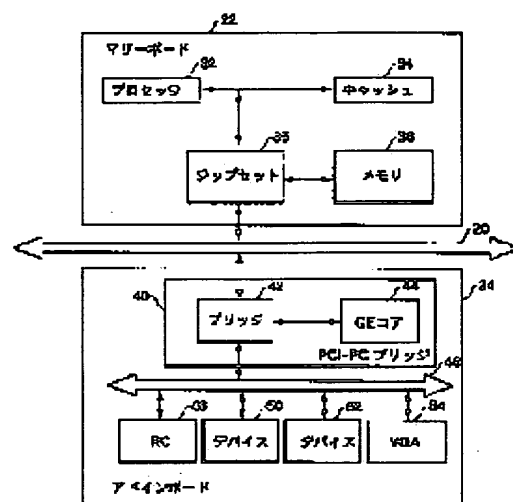
(72)Inventor : ROBERT STREITENBERGER
KAWAI HIROYUKI

(54) PCI-PCI BRIDGE AND ITS FAST-IN FAST-OUT MEMORY

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a customized PCI(peripheral component interconnection)-PCI bridge which is securely supported by existing BIOS and on the chip of which a DMA controller and an LSI core are mounted.

SOLUTION: A customized PCI-PCI bridge 40 identifies a secondary PCI device and realizes its configuration by using a type '00' header and a function number. For supporting a VGA device 54 with an operation core and PCI agent, a memory map is switched over at the time of starting and after starting. Bus driving is mediated so that two PCI buses 20 and 22 are not simultaneously driven.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-238030

(43) 公開日 平成11年(1999) 8月31日

(51) Int.Cl.⁶

G 0 6 F 13/14
13/36

識別記号

3 3 0
3 1 0

F I

G 0 6 F 13/14
13/36

3 3 0 A
3 1 0 E

審査請求 未請求 請求項の数12 O L (全 19 頁)

(21) 出願番号 特願平10-38841

(22) 出願日 平成10年(1998) 2月20日

(71) 出願人 000006013

三菱電機株式会社
東京都千代田区丸の内二丁目2番3号

(72) 発明者 ロバート ストライトテンベルガー
東京都千代田区丸の内二丁目2番3号 三
菱電機株式会社内

(72) 発明者 河合 浩行
東京都千代田区丸の内二丁目2番3号 三
菱電機株式会社内

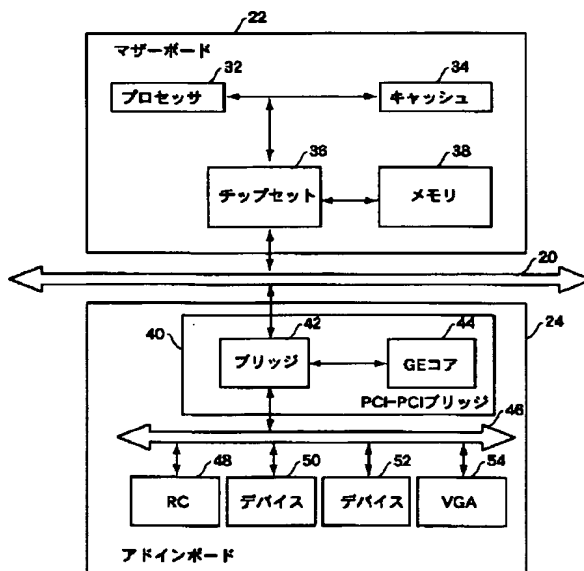
(74) 代理人 弁理士 深見 久郎 (外3名)

(54) 【発明の名称】 P C I - P C I ブリッジおよびそのための先入れ先出しメモリ

(57) 【要約】

【課題】 既存の B I O S により確実にサポートされ、DMA コントローラと L S I コアとを同一チップ上に設けたカスタム化 P C I - P C I ブリッジを提供する。

【解決手段】 カスタム化された P C I - P C I ブリッジ 4 0 はタイプ「0 0」ヘッダを用い、ファンクション番号を用いて 2 次 P C I デバイスを識別しコンフィギュレーションを可能にする。演算コア、P C I エージェントとともに V G A デバイス 5 4 をサポートするために、起動時と起動後とでメモリマップを切替える。また 2 つの P C I バス 2 0、2 2 が同時に駆動されないようにバス駆動を調停する。



20:プライマリPCIバス
46:2次PCIバス

【特許請求の範囲】

【請求項 1】 プライマリ P C I バスと、2 次 P C I バスとに接続される P C I - P C I ブリッジであって、前記 2 次 P C I バスと前記プライマリ P C I バスとを接続し、その間のデータ転送を制御するための接続手段を含み、

前記接続手段は、タイプ「0 0」のコンフィギュレーションヘッダを有し、かつコンフィギュレーション時に、デバイスドライバからのコンフィギュレーションコマンドのファンクション番号フィールドの値に基づいて、前記 2 次 P C I バス上の複数の P C I エージェントを識別してコンフィギュレーションを実行させるための手段を含む、P C I - P C I ブリッジ。

【請求項 2】 前記複数の P C I エージェントは、V G A デバイスと、他の P C I エージェントとを含み、前記接続手段は、V G A デバイスのためのメモリマップのみがアクティブな第 1 のモードと、前記 V G A 以外の P C I エージェントがアクティブな第 2 のモードとの 2 つのうちの 1 つを選択して動作することが可能な、請求項 1 に記載の P C I - P C I ブリッジ。

【請求項 3】 前記接続手段は、起動時には前記第 1 のモードで起動し、前記プライマリ P C I バスに接続された処理手段上で動作するデバイスドライバによって前記第 2 のモードに切り換えられる、請求項 2 に記載の P C I - P C I ブリッジ。

【請求項 4】 前記プライマリ P C I バスおよび前記 2 次 P C I バスとデータ転送が可能で、特定の演算処理を行なうための演算手段をさらに含み、

前記接続手段は、前記プライマリ P C I バスおよび前記 2 次バスと前記演算手段との間のデータの送受を管理する、請求項 1 に記載の P C I - P C I ブリッジ。

【請求項 5】 デバイスドライバによって主記憶からの直接データ転送がトリガされたことに応答して、以後の指定されたデータ転送をデバイスドライバの介在なく行ない、かつ転送されるべきデータブロックの末尾に付されたリンクリストに応答して、前記リンクリストにより新たに定められるデータ転送をデバイスドライバの介在なく行なうためのオンザフライ直接データ転送手段をさらに含む、請求項 1 ~ 4 のいずれかに記載の P C I - P C I ブリッジ。

【請求項 6】 前記接続手段は、前記プライマリ P C I バス上の P C I エージェントおよび前記 2 次 P C I バス上のエージェントからのバス要求に応答し、前記プライマリ P C I バスと前記 2 次 P C I バスとの双方が同時に駆動されないよう、前記プライマリ P C I バスと前記 2 次 P C I バスとの駆動許可を調停制御するための状態機械を含む、請求項 5 に記載の P C I - P C I ブリッジ。

【請求項 7】 前記状態機械が、前記 2 次 P C I バスと同じクロックメインで動作する、請求項 6 に記載の P C I - P C I ブリッジ。

【請求項 8】 前記状態機械は、所定量のデータ転送が終了するごとに状態を遷移すべきか否かを判定し、判定結果に従って状態を遷移させる、請求項 6 または 7 に記載の P C I - P C I ブリッジ。

【請求項 9】 前記状態機械は、特定の状態にあるときは、状態を遷移すべきか否かの判定までに転送されるデータ量を、他の状態にあるときよりも多く設定することが可能である、請求項 8 に記載の P C I - P C I ブリッジ。

【請求項 1 0】 前記オンザフライ直接データ転送手段が、自己によるデータ転送の持続期間を制御するように前記状態機械を設定可能である、請求項 5 ~ 9 のいずれかに記載の P C I - P C I ブリッジ。

【請求項 1 1】 異なるクロック周波数で動作する要素間でのデータの転送をインターフェイスする先入れ先出しメモリであって、

書込ポートのための書込状態機械と、

読出ポートのための読出状態機械と、

前記書込状態機械と前記読出状態機械との間に並列に配置された複数のメモリバンクとを含む、先入れ先出しメモリ。

【請求項 1 2】 請求項 1 1 に記載の F I F O を用いた、請求項 5 ~ 9 のいずれかに記載の P C I - P C I ブリッジ。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】この発明はプライマリ P C I バスと 2 次 P C I バスとの間の P C I - P C I ブリッジに関し、特に、既存の B I O S によってサポートされ、かつ V G A デバイスを含む複数の P C I エージェントをプライマリ P C I を介して制御することを可能にする P C I - P C I ブリッジに関する。

【0 0 0 2】

【従来の技術】近年、特にパーソナルコンピュータでは P C I (Peripheral Component Interconnect) バスの採用が一般的である。P C I は米 Intel 社が提唱したバス規格であって、米国の業界団体である P C I Special Interest Group (P C I S I G) が仕様を策定し公開している。

【0 0 0 3】P C I バス規格では、P C I のデータ転送に参加するデバイスのことを「エージェント」と呼ぶ。P C I 規格では、C P U により実行される B I O S や、デバイスドライバ等によりこれらエージェントが制御される。P C I バス規格では、電源投入時にバスに接続されたデバイスを自動的に感知し、各システムへのメモリマップなどシステム環境を整える自動コンフィギュレーション機能が提供されている。

【0 0 0 4】一方、近年のパーソナルコンピュータの高性能化に伴い、従来はワークステーションで行なわれていた画像処理のプログラムなどをパーソナルコンピュー

タにより実行させることが多くなってきた。そのために、パーソナルコンピュータ内のPCIバスを介してそのような特定の演算処理を行なうLSI（半導体集積回路）をホストCPUに接続することが多くなる。そうした場合には、それら集積回路の中に複数のPCIデバイスを実装することが望ましいことがある。そのときには、ホストのPCIバスと、付加される装置内のPCIバスとの間をインターフェイスする必要が生ずる。このようにホストCPU内のPCIバスをプライマリPCIバス、そのPCIバスに接続される回路（アドインボードと呼ばれる）内のPCIバスを2次PCIバスと呼ぶ。

【0005】プライマリPCIバスに、2次PCIバスを接続し、2次PCIバスに複数のPCIエージェントを接続する場合、ホスト側から各PCIを個別に制御する必要が生ずる。そのために、PCI規格ではPCIコンフィギュレーションヘッダと呼ばれるデータを使用する。このヘッダにはタイプ「00」と「01」との2種類がある。タイプ「00」はホストCPUから見て1つのPCIエージェントとして認識されるものであり、PCI-PCIブリッジについてはタイプ「01」ヘッダが準備されている。

【0006】

【発明が解決しようとする課題】しかしながら、このタイプ「01」によるPCI-PCIブリッジでは、サポートするBIOSが限られていること、DMA（直接メモリアクセス）機能を付加することができないこと、2次PCIバスを持つ装置内にLSIコアを含ませることができないこと、など数々の問題がある。そのためたとえば、1つの半導体集積回路装置内に画像処理のための種々の回路を組み込み、そのうちの1つがPCIエージェントであるような場合には、タイプ「01」ヘッダを使用する不利益が大きかった。また、プライマリPCIバスと2次PCIバスとを介したデータ転送は安定して行なわれなければならない、しかも高速である必要がある。この時、一般にプライマリバスと2次バスとでは2次バスの方が高速である。そのためこの間のデータ転送にはFIFO（先入れ先出しメモリ）を用いるが、仕様変更等があるとFIFOの設計変更を行なう必要が生じ、全体の設計に要する時間が長期化するという問題がある。

【0007】さらにまた、たとえば起動時のトラブルの表示の際には、VGAデバイスを用いた表示が必要とされるなど、既存のVGAデバイスとの互換性を確保することが有利である。一方で、前述したグラフィック処理を行なうためのアドインボードを用いて、たとえば起動時のエラーメッセージの表示などを行なうことも不可能ではないが、そのための処理は既存のVGAデバイスを用いた場合と比較して複雑である。したがってVGAデバイスも含めて2次PCIバスに接続しておき、必要に応じてこれらを切換えることができれば好ましい。この

ようなことはタイプ「01」ヘッダのPCI-PCIブリッジで行なうことは困難なことであった。

【0008】それゆえにこの発明の目的は、タイプ「01」ヘッダを用いずに、2次PCIバスを持つ半導体集積回路装置をプライマリPCIバスに接続し、かつ個別のPCIエージェントを独立に制御できるPCI-PCIブリッジを提供することである。

【0009】この発明の他の目的は、VGAデバイスをPCIエージェントとともに含んだ、2次PCIバスとプライマリPCIバスとの間を接続可能なPCI-PCIブリッジを提供することである。

【0010】この発明の他の目的は、起動時のエラーメッセージの表示を、既存のVGAデバイスを用いて容易に行なうことができるPCI-PCIブリッジを提供することである。

【0011】この発明のさらに他の目的は、DMA転送を可能とするPCI-PCIブリッジを提供することである。

【0012】この発明のさらに追加の目的は、主記憶と、2次PCIバス上のPCIエージェントまたは特定の演算を行なう演算回路との間でのデータ転送を高速で行なうことができるPCI-PCIブリッジを提供することである。

【0013】この発明のさらに追加の目的は、PCIブリッジで用いられ、PCI-PCIブリッジの設計を容易にすることができるFIFOを提供することである。

【0014】

【課題を解決するための手段】請求項1に記載の発明にかかるPCI-PCIブリッジは、プライマリPCIバスと、2次PCIバスとに接続されるPCI-PCIブリッジであって、2次PCIバスとプライマリPCIバスとを接続し、その間のデータ転送を制御するための接続手段を含む。接続手段は、タイプ「00」のコンフィギュレーションヘッダを有し、かつコンフィギュレーション時に、デバイスドライバからのコンフィギュレーションコマンドのファンクション番号フィールドの値に基づいて、2次PCIバス上の複数のPCIエージェントを識別してコンフィギュレーションを実行させるための手段を含む。

【0015】タイプ「00」のヘッダを使用するので、このPCI-PCIブリッジはBIOSにより確実にサポートされる。また、BIOSからPCI-PCIブリッジとは認識されないが、コンフィギュレーションヘッダのファンクション番号フィールドを用いて、2次PCIバス上の複数のPCIエージェントを識別して個々にコンフィギュレーションでき、デバイスドライバにより2次PCIバス上のエージェントを柔軟に制御することができる。

【0016】請求項2に記載の発明にかかるPCI-PCIブリッジは、請求項1に記載の発明の構成に加え

て、複数のPCIエージェントは、VGAデバイスと、他のPCIエージェントとを含む。接続手段は、VGAデバイスのためのメモリマップのみがアクティブな第1のモードと、VGA以外のPCIエージェントがアクティブな第2のモードとの2つのうちの1つを選択して動作することが可能である。

【0017】VGAデバイスのみが動作可能な第1のモードと、VGA以外のPCIエージェントが動作可能な第2のモードとを切り換えて使用することができるので、VGAを用いる、従来のDOS上で動作するプログラムと、PCIエージェントによる新規な作業とを共に実行することができる。

【0018】請求項3に記載の発明にかかるPCI-PCIブリッジは、請求項2に記載の発明の構成に加えて、接続手段は、起動時には第1のモードで起動し、プライマリPCIバスに接続された処理手段上で動作するデバイスドライバによって第2のモードに切換えられる。

【0019】起動時にVGAのためのメモリマップのみがアクティブな第1のモードで立ち上がるので、たとえばエラーメッセージの表示などをVGAを用いて容易に行なうことができる。また一旦デバイスドライバが動作し始めると第2のモードに切換られるので、PCIデバイスを用いた処理を円滑に実行開始できる。

【0020】請求項4に記載の発明にかかるPCI-PCIブリッジは、請求項1に記載の発明の構成に加えて、プライマリPCIバスおよび2次PCIバスとデータ転送が可能で、特定の演算処理を行なうための演算手段をさらに含む。接続手段は、プライマリPCIバスおよび2次バスと演算手段との間のデータの送受を管理する。

【0021】演算手段をPCI-PCIブリッジ上を含むことにより、ホストCPUに対して強力なサービスを提供できる。特に、グラフィック処理のように計算量が多い場合、そのための専用の演算手段を設けることでホストCPUの負荷を減じることができる。

【0022】請求項5に記載の発明にかかるPCI-PCIブリッジは、請求項1～4のいずれかに記載の発明の構成に加えて、デバイスドライバによって主記憶からの直接データ転送がトリガされたことに応答して、以後の指定されたデータ転送をデバイスドライバの介在なく行ない、かつ転送されるべきデータブロックの末尾に付されたリンクリストに応答して、リンクリストにより新たに定められるデータ転送をデバイスドライバの介在なく行なうためのオンザフライ直接データ転送手段をさらに含む。

【0023】最初のDMA転送のセットアップさえ終われば、以後はオンザフライでDMA転送のコンフィギュレーションが行なわれ、次のDMA転送が自動的に行なわれるので、主記憶から他へのデータ転送において、ホ

スト（デバイスドライバ）のオーバーヘッドが少なく、効率的なデータ転送を行なうことができる。

【0024】請求項6に記載の発明にかかるPCI-PCIブリッジは、請求項5に記載の発明の構成に加えて、接続手段は、プライマリPCIバス上のPCIエージェントおよび2次PCIバス上のエージェントからのバス要求に応答し、プライマリPCIバスと2次PCIバスとの双方が同時に駆動されないよう、プライマリPCIバスと2次PCIバスとの駆動許可を調停制御するための状態機械を含む。

【0025】プライマリPCIバスと2次PCIバスとが同時にスイッチすると、それらによるPCI-PCIブリッジの動作電圧の変動によって、PCI-PCIブリッジが誤動作をする危険がある。しかし、状態機械によってこれら二つのバスが同時に駆動されないようにバスの駆動許可を調停するので、こうしたおそれは少なくなる。

【0026】請求項7に記載の発明にかかるPCI-PCIブリッジは、請求項6に記載の発明の構成に加えて、状態機械が、2次PCIバスと同じクロックドメインで動作する。

【0027】2次PCIバスに対するデータ転送が主に行なわれる場合、SSOが2次PCIバスと同じクロックドメインで動作すると、そうでない場合と比較して効率的なデータ転送が可能になる。

【0028】請求項8に記載の発明にかかるPCI-PCIブリッジは、請求項6または7に記載の発明の構成に加えて、状態機械は、所定量のデータ転送が終了するごとに状態を遷移すべきか否かを判定し、判定結果に従って状態を遷移させる。

【0029】全てのデータソースからのデータ転送が公平に行なわれるために、状態機械による状態遷移を、一定の期間ごとに行なわせるようにすることが望ましい。一定期間ごとに状態機械の状態が遷移することにより、全てのデータソースからのデータ転送が同様の効率で行なわれる。

【0030】請求項9に記載の発明にかかるPCI-PCIブリッジは、請求項8に記載の発明の構成に加えて、状態機械は、特定の状態にあるときは、状態を遷移すべきか否かの判定までに転送されるデータ量を、他の状態にあるときよりも多く設定することが可能である。

【0031】ある種のデータ転送では、何回かに分けてデータ転送を行なうと、たとえば同期ペナルティによりデータ転送の効率が低下する。これは、装置全体のパフォーマンスの低下をもたらす。そのため、このようなデータ転送では、状態機械の状態が遷移するまでに行なえるデータ転送の量を、他の場合と比較して多くする。こうすることでデータ転送を分けるときのパフォーマンスの低下を避けることができ、装置全体のパフォーマンスが向上する。

【0032】請求項10に記載の発明にかかるPCI-PCIブリッジは、請求項5～9のいずれかに記載の発明の構成に加えて、オンザフライ直接データ転送手段が、自己によるデータ転送の持続期間を制御するように状態機械を設定可能である。

【0033】オンザフライ直接データ転送手段によるデータ転送も、分割することにより効率が低下することがある。そこで、たとえば転送データ量が多い場合にはデータ転送の持続期間を長く、そうでない場合には比較的短く設定できるように状態の遷移を柔軟に制御できるようにすることで、装置全体としてのパフォーマンスが向上する。

【0034】請求項11に記載の発明にかかる先入れ先出しメモリは、異なるクロック周波数で動作する要素間でのデータの転送をインターフェイスする先入れ先出しメモリであって、書込ポートのための書込状態機械と、読出ポートのための読出状態機械と、書込状態機械と読出状態機械との間に並列に配置された複数のメモリバンクとを含む。

【0035】書込状態機械と、読出状態機械との間に並列に配置するメモリバンクの数によって、先入れ先出しメモリの大きさを容易に変更できる。先入れ先出しメモリをこのように構成することで、たとえばハードウェア記述言語などによる設計では、先入れ先出しメモリの大きさを、配置されるメモリバンクの数という一つのパラメータに還元することができる。そのため仕様変更や、突然の設計変更があっても容易に先入れ先出しメモリの設計を修正することができ、装置の開発期間を短くすることができる。

【0036】請求項12に記載の発明にかかるPCI-PCIブリッジは、請求項11に記載のFIFOを用いた、請求項5～9のいずれかに記載のPCI-PCIブリッジである。

【0037】このPCI-PCIブリッジでは、データ転送を高速に、効率よく行なえる上、データの転送の際に用いられる先入れ先出しメモリの大きさを容易に変更できるため、仕様変更や、突然の設計変更があってもPCI-PCIブリッジの設計を簡単に修正することができ、装置の開発期間を短くすることができる。

【0038】

【発明の実施の形態】以下、グラフィックプロセッサに組み込まれたPCI-PCIブリッジを使いアドインボードとして実現した実施の形態について説明する。

【0039】[全体構成]図1を参照して、この発明に係るPCI-PCIブリッジの一実施の形態に係るPCI-PCIブリッジ40は、アドインボード24上に実装され、プライマリPCIバス20に接続されている。プライマリPCIバス20にはマザーボード22が接続されている。このマザーボード22上のプロセッサ32で実行されるBIOSやデバイスドライバなどによって

アドインボード24上の各回路が制御される。

【0040】マザーボード22は、プロセッサ32、キャッシュ34、メモリ38、ならびにプロセッサ32、メモリ38およびプライマリPCIバス20の間のデータ転送を制御するチップセット36を含む。

【0041】アドインボード24は、PCI-PCTBブリッジ40によってプライマリPCIバス20と接続される2次PCIバス46と、2次PCIバス46にそれぞれ接続されたグラフィックレンダリングのためのRC（レンダリングコントローラ）48、2種類のデバイス50および52、およびVGAデバイス54とを含んでいる。PCI-PCIブリッジ40は、画像処理のうち幾何学的演算を行なうためのGE（Geometric Engine）コア44と、プライマリPCIバス20、2次PCIバス46およびGEコア44の間のデータ転送をインターフェイスするためのブリッジ部42を含む。なお、PCIデバイスの構成例がこの実施の形態の装置で示したものに限定されないのはもちろんである。

【0042】図2を参照して、PCI-PCIブリッジ40をさらに詳細に説明すると、PCI-PCIブリッジ40はブリッジ部42とGEコア44との間のコアI/O66と、ブリッジ部42とプライマリPCIバス20との間のプライマリI/O62と、ブリッジ部42と2次PCIバス46との間の2次I/O64とを含んでいる。GEコア44により、画像処理の強力なサポートが得られ、プロセッサ32の負荷が軽減される。

【0043】プライマリI/O62は、ターゲットFSM（有限状態機械）72およびマスタFSM74を含む。また2次I/O64は2次PCIバス46との間の入出力を制御するマスタFSM76を含む。2次I/O64は、後述するようにプライマリPCIバス20と2次PCIバス46との動作を制御する状態機械である図示しないSSO（Simultaneous Switching Outputs）を含んでいる。

【0044】ブリッジ部42は、プライマリI/O62と2次I/O64の間のFIFO80と、それぞれプライマリI/O62とGEコア44との間、2次I/O64とGEコア44との間に設けられたFIFO84および86とを含んでいる。

【0045】この実施の形態のアドインボード24では、PCIバス規格のコンフィギュレーション領域ヘッダのうち、タイプ「00」ヘッダを用いており、全てのBIOSで確実にサポートされる。この場合タイプ「01」コンフィギュレーション領域による時のようなPCI-PCIブリッジのためのBIOSによるサポートがないため、以下に述べるように本実施の形態では特別な方法を用いている。

【0046】タイプ「00」ヘッダを用いているので、アドインボード24内のコンフィギュレーション領域は、このブリッジのチップを表わしているだけである。

その中に含まれる各 P C I エージェントなどはプロセッサ 3 2 で実行される B I O S から見ることはできない。したがって起動時に B I O S の一部でありコンフィギュレーションのための情報を取得する P O S T ルーチン実行時に、アドインボード 2 4 内の各 P C I エージェントや V G A などはメモリ空間を B I O S に対して要求することができない。したがってこの実施の形態のアドインボード 2 4 では、P C I - P C I ブリッジ 4 0 がそれら P C I エージェントなどに代わってメモリ領域を要求する。ここでこの実施の形態のアドインボード 2 4 では V G A デバイス 5 4 が備えられているため、起動時にはこれ以外にメモリ領域の管理について問題があるが、それについては後述する。

【0047】一意の I D を与える I D レジスタ（ベンダ／デバイスおよびリビジョン I D など）以外のすべてのレジスタは、プライマリ P C I バスを介してアドインボード 2 4 がどのように見えるかを表わしている。たとえばレジスタ 0 4 h のステータス／コマンドビットはプライマリ P C I バス上でのアドインボードの動作を決定する。ブリッジ内のコンフィギュレーション領域のレジスタ 0 4 h のビット [1] によってメモリサイクルがディスエーブルされている場合、コンフィギュレーションの設定によればメモリサイクルを得ることができるような場合であっても、2 次 P C I バス上のデバイスにはデータは転送されない。

【0048】初期化／リセット直後にはアドインボードはディスエーブル状態である。P O S T ルーチンの間に、B I O S によってブリッジ自体に対しメモリ空間が割当てられる。このときの割当てはブリッジのコンフィギュレーション領域のレジスタ 1 0 h の値を参照して行なわれる。このメモリ空間の割当て要求はアドインボードを代表して P C I - P C I ブリッジ 4 0 が行なうものである。B I O S はこの空間を P C I - P C I ブリッジ 4 0 に対して割当てが、この領域内をどのように区分するかについては関与しない。領域をどのように分割するかは、P C I - P C I ブリッジ 4 0 とプロセッサ 3 2 上で動作する、デバイスドライバとによってのみ決定される。

【0049】この実施の形態の P C I - P C I ブリッジではタイプ「0 1」ヘッダを使用していないので、B I O S は 2 次 P C I バスの存在を関知しない。したがってこの場合、B I O S に代わってデバイスドライバが 2 次 P C I バス上のデバイスを初期化し、コンフィギュアしなければならない。その作業として第 1 に挙げられるのは、ベースアドレスレジスタ（B A R）の設定である。これは、各デバイスに割当てられるメモリ空間のオフセットを表わす。各デバイスへのメモリ空間がそれらに対して割当てられているわけではないので、「メモリ読出／書込」コマンドを用いてはこれらにアクセスすることができない。しかしデバイスドライバは、特別な「コン

フィギュレーション読出／書込」コマンド（以下に説明する）を用いてこれらデバイスのコンフィギュレーション領域にアクセスすることができる。

【0050】ブリッジのコンフィギュレーション領域のレジスタ 1 0 h 内に B I O S が書込んだこの値に基づいて、デバイスドライバがメモリマップに従い 2 次 P C I バスに接続されているデバイスのベースアドレスを割当てる。したがってデバイスドライバは、2 次 P C I バス上の各デバイスのコンフィギュレーション領域のレジスタ 1 0 h にそれぞれ対応の値を書込まなければならない。

【0051】P C I 仕様書（リビジョン 2. 1）に記載されているコンフィギュレーション機構 # 1 は、ホストブリッジを介してコンフィギュレーションサイクルを生成するために 2 つの D W o r d（D W o r d とは 2 つの 1 6 ビットワードを指す）I / O ロケーションを用いる。1 番目の D W o r d ロケーション（C F 8 h）は「C O N F I G _ A D D R E S S」と呼ばれる R / W レジスタを指す。このレジスタ 9 0 のフォーマットを図 3 に示す。

【0052】図 3 を参照してこのレジスタ 9 0 は、ターゲットのコンフィギュレーション領域のレジスタ番号（ビット [7 : 2]）およびファンクション番号（ビット [1 0 : 8]）を含んでいる。タイプ「0 0」ヘッダを用いているため、このブリッジはマルチファンクションのデバイスとはみなされておらず、ファンクション番号の 3 ビットは B I O S によっては使用されない。そのためデバイスドライバはこの 3 ビットを自由に使用することができる。この 3 ビットを用いてターゲットのコンフィギュレーション領域を決定するが、その態様を図 4 に表形式で示す。

【0053】図 4 において「A D」はアドレス・デバイスバス上のデータを指す。「Destination」は宛先を指す。また図 4 の最も右側の欄は、この宛先に対応するターゲットとなるデバイス例を示す。図 4 に示す例のうち、V G A デバイスが存在する場合には必ずこのファンクション番号が割当てられる。しかしそれ以外のデバイスに対する経路制御はシステム設計により任意に定められる。

【0054】ブリッジのコンフィギュレーション領域はファンクション番号を 0 0 0 b（または暗黙値である 1 0 1 b、1 1 0 b、1 1 1 b）に設定することにより直接にかつレイテンシなくアクセスすることができる。たとえばデバイス 1（D e v _ 1）をアクセスする場合にはファンクション番号は 0 0 1 b に設定しなければならない。プライマリ P C I バス上においてブリッジによりコンフィギュレーションサイクルが生成され取得される。なぜならばこのとき「I D S E L」が活性化されるためである。

【0055】この信号はいわゆるチップセレクト信号に

相当するものであって、コンフィギュレーションを行なうデバイスを選択するためのものである。このアドレスは、ブリッジ部 4 2 内のプライマリ F S M (Finite State Machine: 有限状態マシン) により分析される。ファンクション番号ビットに「1 0 0 b」があった場合ブリッジ内の「Bridge Data Path」(図 2 においてプライマリ 1 / O 6 2 と 2 次 1 / O 6 4 とを結んでいる) を使用してこのサイクルが P C I - P C I ブリッジ内の 2 次 F S M に伝搬される。図 4 の表に従って、対応する「S _ A _ D」(2 次アドレス・データ) ラインが活性化され、2 次 P C I バス上にこの伝搬されたコンフィギュレーションサイクルが生成される。この場合ファンクション番号は 0 0 0 b にリセットされる。

【0 0 5 6】[VGA] VGA は D O S 互換の、固定されたメモリ領域 (A 0 0 0 0 - B F F F F h) と、固定された I / O ポート群 (3 B 0 - 3 B B, 3 C 0 - 3 D F h) とを有している。しかし P C I バス上の VGA デバイスはまた、メモリおよび I / O 領域に関しリロケート可能となっている。

【0 0 5 7】システム内に複数の VGA デバイスが存在する場合、それら VGA デバイスは同一のメモリおよび I / O 領域を共有することができない。そのため B I O S はこれら VGA デバイスをイネーブルする前に必要に応じて各メモリ領域をリロケートする。すべての P C I デバイスはコンフリクトを避けるために、P O S T ルーチンが終了するまでは、起動時にはメモリおよび I / O 領域をディスエーブルさせておく必要がある。

【0 0 5 8】本実施の形態の P C I - P C I ブリッジは D O S 互換のメモリ領域とリロケートされたメモリ領域との双方をサポートしている。しかしここでサポートされているのは、下位 6 4 K B および特定のアドレス内での D O S 互換のデフォルトの I / O 領域とそのリロケートされたものだけである。

【0 0 5 9】この P C I - P C I ブリッジではタイプ「0 0」ヘッダを用いている。そのため 2 次 P C I バス上のデバイスは B I O S から見ることはできない。2 次 P C I バス上に VGA デバイスが存在している場合、したがって P O S T ルーチン実行時に B I O S に対して V G A デバイスが存在していることを示さなければならない。

【0 0 6 0】ここでこの実施の形態の P C I - P C I ブリッジは、2 次 P C I バス上に V G A デバイスが存在しているかどうかを予め知っておく必要がある。この情報は、たとえば図 5 に示されるように或る特定のピンに対して与えられるストラップ信号により設定される。この信号はたとえばディップスイッチのようなもので与えられるものであって、2 次 P C I ブリッジに接続されるデバイスが V G A デバイスを含むか否かに従って 1 回だけ設定されるものである。図 5 に示す例では、このピンに与えられる信号が 0 b である場合には V G A デバイスが

存在していないことを、1 b である場合には V G A デバイスが存在していることを、それぞれ示す。

【0 0 6 1】このストラップ信号の値が「オン」である場合、すなわちピン上の値が 1 b である場合には、コンフィギュレーション領域内の V G A に関するレジスタについては、P C I - P C I ブリッジによってシャドーされなければならない。この場合の「シャドー」とは、簡単に言えば複製を保持しておくということである。すなわちこの場合、デフォルトのレジスタに加えて V G A に関するアクセス専用のレジスタも P C I - P C I ブリッジ内に用意されることになる。ストラップ信号の値に応じて、P C I - P C I ブリッジのコンフィギュレーション領域に対する読出／書込アクセスに対し、デフォルトのレジスタと V G A 専用レジスタとのいずれかが選択される。

【0 0 6 2】シャドーされるレジスタとしてはコマンドレジスタ [0 4 h]、アドインボードメモリベースアドレスレジスタ [1 0 h]、および拡張 R O M ベースアドレスレジスタ [3 0 h] がある。

【0 0 6 3】図 6 を参照して、P C I - P C I ブリッジ 4 0 内には、V G A デバイス 5 4 のレジスタ 0 4 h のコピーが保持されている。ストラップ信号がオンのとき、コンフィギュレーション読出アクセスのときには図 6 に示されるように V G A デバイス 5 4 のレジスタ 0 4 h からでなく、P C I - P C I ブリッジ 4 0 のレジスタ 0 4 h に対応する複製からその値が読出される。また書込のときには、図 7 に示されるように P C I - P C I ブリッジ 4 0 の複製レジスタに書込が行なわれるとともに、P C I - P C I ブリッジ 4 0 内でこのサイクルに対するマスクが行なわれ、2 次 P C I バス 4 6 を介して V G A デバイス 5 4 のコンフィギュレーション領域内のレジスタ 0 4 h に対して書込が行なわれる。

【0 0 6 4】したがって、V G A デバイスのコンフィギュレーション領域のうちの主要なレジスタについては、P O S T の間に必要に応じて初期化が行なわれることになる。B I O S はコマンドレジスタ内の対応ビットをセットすることによってメモリおよび I / O 領域をイネーブルし、この情報は V G A デバイスにも与えられ、したがって V G A デバイスの初期化／コンフィギュレーションが実行されることになる。

【0 0 6 5】ここで、注意しなければならないのは、V G A デバイスに関するコンフィギュレーションサイクルを 2 次 P C I バス上の V G A デバイスに与える場合種々の制約があるということである。たとえばこの P C I - P C I ブリッジは V G A デバイスについての D A C (デジタル／アナログ変換) パレットのスヌープをサポートしていない。したがって P C I - P C I ブリッジのコンフィギュレーション領域のビット [5] は「0」に結線され書込可能ではない。仮にコンフィギュレーションサイクルを完全な形で 2 次 P C I バスに伝搬した場合、

VGAデバイスのコンフィギュレーション領域のレジスタ04hに誤って書込されることがあり、2次PCIバス上でのスヌーピングが可能化されてしまうという不都合がある。

【0066】そこで、このように誤ってレジスタへの書込を行なわないように特定のビット領域についてはマスクすることが行なわれる。たとえばレジスタ04hの場合には図8に示されるようにビット[15:2]が強制的に「0」とされる。

【0067】このように選択的にマスクすることによりステータスビットがクリアされVGAデバイスのメモリおよびI/O領域が可能化される。ステータスビットをクリアするのは、リセット後のステータスビットがクリアされていない場合にはVGAデバイスが誤動作する恐れがあるため、その予防である。BIOSは通常は、VGAデバイスに対する他のアクセスが発生する前に初期化段階でこれらビットをクリアする。

【0068】レジスタ10hは読出/書込許可に関する限りはストラップには関係しない。しかしこのレジスタに対するコンフィギュレーション書込アクセスが発生した場合にはこのアクセスは2次側に伝搬される。その伝搬時のマスキングを図9に示す。

【0069】図9に示されるように下位27ビットをマスクすることにより、VGAデバイスのメモリ領域が128Mバイトに整列される。すなわち、VGAデバイスのメモリ領域の下部境界が、PCI-PCIブリッジが要求したメモリ領域の最下部と一致する。

【0070】レジスタ30hは完全にストラップに従属するものであり、ストラップ信号がオフの場合には書込することができない。ストラップがオンの場合には図10に示されるようなマスキングが行なわれる。下位24ビット(LSBが「0」であるものとする)をマスキングすることによって、VGAデバイスのROM領域が16Mバイトに整列されることになり、かつLSBを操作することによりBIOSが拡張ROMをイネーブル/ディスエーブルすることができる。

【0071】VGAデバイスのコンフィギュレーション領域内のレジスタ10hによって要求されるメモリ領域のサイズは各VGAデバイスによって異なる。そのサイズは数Mビットから、大きい場合には64Mバイトまたはそれ以上ともなり得る。可能な限り多くの種類のVGAデバイスをサポートするために、次のように特別な手段を採用する。

【0072】すなわち、デバイスドライバが、図11に示されるような一般的メモリマップと、図12に示されるようなVGAメモリマップとの間で切換をすることができるようになる。この切換は、PCI-PCIブリッジのコンフィギュレーション領域のレジスタ40h内のビット[0]を操作することにより切換えることができる。図11を参照して、一般的メモリマップでは、メモ

リ領域120は、下部4Mバイトのブリッジ(制御)のための領域132と、その上の4Mバイトのブリッジ

(コア)のための領域134と、各デバイスのための領域136とに分けられる。一方、VGAメモリマップでは、メモリ領域120は124MバイトのVGA領域138と、その上のブリッジ(制御)のための領域140(4Mバイト)とに分けられる。

【0073】起動時にはPCI-PCIブリッジは、VGAメモリマップをアクティブにして立上がる。BIOSが5ビットオフセットを割当て、すなわち128Mバイトのメモリ領域を割当てた後、PCI-PCIブリッジは下位124Mバイトを2次PCIバスに伝搬する。VGAデバイス以外の他のデバイスはすべてメモリおよびI/O領域をディスエーブルとして立上がるので、BIOSによりイネーブルされたVGAのみがサイクルを取得する。ブートアップが完了するとデバイスドライバが機能を開始する。デバイスドライバはVGAをディスエーブルし他のデバイスをイネーブルして一般的メモリマップに切換える。なお、このように起動時にVGAメモリマップで立上げを行なうのは、何らかのエラーが発生した場合、エラーメッセージをVGAデバイスを用いて表示することが簡便であるためである。またPCI-PCIブリッジのコンフィギュレーション領域のレジスタ40hのビット[0]を操作することにより図11のメモリマップと図12のメモリマップとを切換えることができる。そのためたとえば従来のDOS互換のプログラムを使用する場合には図12に示されるメモリマップに切換えてそれらのプログラムを実行することができる。

【0074】エラー処理をサポートするためには、その要求を処理するドライバをイネーブルするために、適切な応答を選択し、リソースをイネーブルし、ステータスフラグがアクセス可能とならなければならない。仮にPCI-PCIブリッジメモリ領域がイネーブルされていなければ、すなわちメモリマップされたコンフィギュレーション領域レジスタがアクセスできなければこのようなことはできない。したがって一般的メモリマップの下部4Mバイトに存在していたPCI-PCIブリッジメモリ領域を、VGAメモリマップの上部4Mバイトにリロケートすることができるようにした。

【0075】VGA関連のコンフィギュレーションサイクルをVGAデバイスに伝搬するために、図13に示されるようにVGAデバイス54の「IDSEL」ラインを一定のラインに接続しておかなければならない。この接続関係は図4の表に示すとおりである。

【0076】[DMAアクセス] DMA(直接メモリアクセス)を実行する場合には、PCI-PCIブリッジはプライマリPCIバスに関して主メモリ、すなわち「ホストブリッジ」(チップセット内に存在する)をターゲットとするマスタとして動作する。データはGEコ

ア44へ、または2次PCIバス46へ転送される。

【0077】DMAアクセスを開始させるためには、デバイスドライバはPCI-PCIブリッジ内のDMA関連のレジスタに値をロードし、DMAアクセスをトリガするフラグをセットしなければならない。転送されるブロックの開始アドレスはレジスタ'DMA_Source'に格納される。このアドレスは主記憶のメモリ領域内の有効なアドレスでなければならない。すなわちプライマリPCIバスにおいて、PCI-PCIブリッジとは異なるターゲットが応答しなければならない。DMAアクセスの実際のターゲットはレジスタ'DMA_Destination'にロードされるアドレスによって決定される。このアドレスは図11に示すようなメモリマップに従って検査され、その結果に従ってPCI-PCIブリッジに、すなわち2次PCIバスにデータが転送される。

【0078】転送されるデータの量はレジスタ'DMA_DWords'に、ブロックを構成するDWordの数をロードすることにより決定される。

【0079】さらに、前述したとおりDMAアクセスを開始するためには、フラグ'DMA_Trigger'をセットする必要がある。

【0080】上述したDMA関連のレジスタのうちレジスタ'DMA_Source'および'DMA_Destination'は順次インクリメントされ、現在のソースおよびターゲットアドレスに対するポインタとしての役目を果たす。レジスタ'DMA_DWords'はインクリメントされず、DMA転送のために用意された専用のカウンタの値と比較が行なわれる。仮にターゲットアドレスがメモリマップ内に存在していない場合には、PCI-PCIブリッジはアイドルとなり、DMA転送は発生しない。

【0081】プライマリPCIバス上でのDMA転送が終了した場合、デバイスドライバに対してその終了が伝えられなければならない。この場合2つの方法を選択することができる。

【0082】プライマリPCIバス上での転送が終了すると、メインメモリ(MM)PCIレジスタのうちの1つの或るビットが必ずセットされ、それによって割込要求が設定される。しかしこの割込要求は他のコントロールレジスタによってマスクすることができる。

【0083】割込による方法はそれほど効率的ではないので、他のMM PCIレジスタ'DMA_Semaphore'を使ってデバイスドライバに通知する方法をとることができる。デバイスドライバは、初期化後1回だけ、DMA転送の前にこのレジスタに主メモリ内のアドレスを書込む。主メモリ内のこのロケーションはデバイスドライバによってアクセスすることができDMA転送の終了を示すセマフォとして使用される。デバイスドライバはセマフォを初期化するためにオールゼロをこの

ロケーションに書込む。プライマリPCIバスでの転送が終了すると、DMA FSMは、主メモリ内に存在するセマフォをターゲットとするメモリ書込サイクルを発行し、MM PCIレジスタ'Global_Status'を転送する。これによりセマフォのLSBは「1」にセットされ、ドライバに対してDMA転送が完了したことを通知する。

【0084】DMA転送の終了を通知するのにこのように1ビットを使用するのに加えて、セマフォの残りの31ビットによってPCI-PCIブリッジの完全な「スナップショット」が提供される。これによってデバイスドライバはPCI-PCIブリッジの状態をオンザフライで、かつオーバーヘッドなく分析することが可能になる。

【0085】DMAバーストの長さは以下によって決定される。

(1) 転送されるべきDWordの数。すなわちレジスタ'DMA_DWords'に書込まれた値。

【0086】(2) ターゲットからの応答。すなわち切断する旨の応答。

(3) レイテンシタイマの値。すなわちPCI-PCIブリッジのコンフィギュレーション領域のレジスタ0Chの値。

【0087】(4) ポスティングバッファのステータス。

ポスティングバッファが一杯である場合、マスタはランザクションを停止してバッファがデータを受取り可能となるまで待ち、バッファがデータ受取り可能となると新たなサイクルを開始する。PCI-PCIブリッジのレイテンシタイマが満了し、プライマリPCIバスのハンドシェイキングのための信号'P_GNT'がデアサートされた場合も同様である。デバイスドライバは起動時にレイテンシタイマの値をチェックし、必要であれば異なる値を書込むこともできる。

【0088】レイテンシカウンタは各サイクルごとに、データ転送がなくともインクリメントされる。すなわち、ターゲットからの応答とは独立にレイテンシカウンタはカウントアップされる。レイテンシカウンタの値がレジスタ'Latency_Timer'に等しくなると信号'P_FRAME'はデアサートされマスタが終了する。

【0089】プライマリPCIバス上でマスタアポートエラーが発生するとフラグ'DMA_Trigger'はリセットされ、DMA転送は終了され、対応のエラー処理が実行される。これは実際のDMA転送の場合もセマフォ書込の場合も同様である。

【0090】DMAは常にポスティングバッファを介して実行される。DMA FSMはMM PCIレジスタ'FIFO_Control'を、ポスティングバッファをアクセスする前にチェックすることはしない。

【0091】本実施の形態のPCI-PCIブリッジは、DMA性能を向上させるために、DMAマスタに対して「リンク」オプションを追加している。このオプションによって、複数のDMA転送の要求を発するというオーバーヘッドなしに、デバイスドライバが複数のDMA転送を行なうことができる。

【0092】このためにデバイスドライバは、DMAトランザクションをトリガする前に、主メモリ内のデータブロックの間にリンクデータを挿入する必要がある。リンクデータはアドレスおよび制御情報を含んだ3つのDWordからなっており、対応のMMIO PCIレジスタに書込まれる。これらDWordとMMIO PCIレジスタとの関係を図14に示す。

【0093】図15を参照して、デバイスドライバがDMAデータをセットアップした後の主メモリでは、たとえばデータブロックnの直後に3つのDWordからなるリンクデータブロックが挿入されている。このリンクデータの後に次のブロックが続くことになる。このデータブロックnの最後に付加されたリンクデータ内の'DMA_Destination'により、データブロックn+1のデータの転送先が定められる。

【0094】このようにリンクデータを使って複数のデータ転送を自動的に実行させるか否かは、MMIO PCIレジスタ'DMA_Control'のビット[29]('DMA_Continue')によりイネーブルまたはディスエーブルすることができる。このビットは、転送されるデータブロックの最後にリンクデータが付加されているか否かを示す。このビットが「1」にセットされていればDMA FSMはこの後にリンクデータが続くものと想定し、その結果次の3つのDWordがMMIO PCIレジスタに書込まれる。こうして最新のリンクデータ('DMA_Control')によってオンザフライで再コンフィギュアした後、DMAマスタはサイクルを終了することなく次のデータブロックを読もうとする。仮に'DMA_Continue'が0にリセットされている場合、データブロックの最後のDWordが転送された後にトランザクションは終了されることになる。

【0095】このようにして、DMA転送の完了ごとに、自動的に新たなDMAトランザクションが開始される。このためにはデバイスドライバは主メモリにリンクリストを一度だけ書込めばよく、それによって以後はPCI-PCIブリッジにより複数個のDMA転送がデバイスドライバからの指示なく実行可能である。

【0096】仮にリンクリストのDMAトランスファの間に、たとえば長時間にわたる転送を検証するために、セマフォを更新する必要がある場合にはそのためのビットをセットすればよい。

【0097】図16にこの方式の原理を示す。デバイスドライバは最初に、DMAマスタに対し4つのデータD

Wordを転送するように設定を行ない、さらにその後リンクデータを読むようにさせる(a1)。それに続いてDMA FSMがリンクデータによってコンフィギュアされ、次の4つのDWordを受け、さらに処理を続ける(a2)。2番目のリンクデータによってDMAマスタは8つのDWordを受けると設定され、しかる後にDMAトランザクションを終了しセマフォを書く(a3)。8つのDWordが転送されセマフォが書込まれた後でDMA FSMはIDLE(アイドル)状態に戻る。

【0098】図17に、ネストされたセマフォ書込を含んだDMAリンクの方式について示す。図16に示したのと同様にデバイスドライバはまず、DMAマスタがデータの4つのDWordを転送しさらにその後引き続きDMA転送を実行させるようにする。ただしこの場合デバイスドライバはセマフォオプションをイネーブルするものとする。その結果4つのDWordが読まれセマフォが書かれた後DMAマスタは処理を続行する(b2)。これによってDMAマスタはリンクデータに従ってさらに4つのDWordを読み込み、さらに処理を続行することになる(b2)。この場合にはセマフォオプションがイネーブルされていないためPCI-PCIブリッジはセマフォの書込を行なわない。そしてさらに8つのDWordを受けてDMAトランザクションを終了しセマフォを書込む(b3)。

【0099】DMAトランザクションの効率を左右する要因として3つが考えられる。

(1) PCI-PCIブリッジにおいて、後述するSSOアービタの制約により、プライマリPCIバス上において新たなアドレスフェイズが開始されると、2次側ではデータを取り出すことができない。しかしアドレスフェイズが終了しデータフェイズとなれば2次側からデータを取り出すことができる。したがって仮にDMAトランザクションが非常に長い場合、アドレスフェイズとしては最初の1回しかないのでDMAトランザクションの効率は非常に良い。なぜなら2次側は、プライマリ側がシステムメモリからデータを読み出すのと並行にデータを取り出すことができるためである。この場合転送に対して影響するのは最初のレイテンシのみである(SSOペナルティと呼ぶ)。

【0100】(2) さらに、プライマリPCIバスアービタがトランザクション後またはトランザクション中にバス使用の許可を奪った場合には、PCI-PCIブリッジは再度プライマリバスの使用許可を要求しなければならない(再調停ペナルティ)。

【0101】(3) 多くのホストブリッジでは最初のレイテンシが非常に長い(16サイクルに達することもある)。したがって仮にDMAトランザクションが早く終了した場合、PCI-PCIブリッジはSSOアービタとプライマリバスアービタ遅延に対処するだけでな

く、さらに加えてホストブリッジによる最初のアクセス遅延にも対処しなければならない。

【0102】そのために、次に説明するような「ドミノ」と呼ばれる機能を追加した。この機能では、システムメモリからの連続したメモリ読出バーストを、ホストブリッジがトランザクションを終了させるまで行なうことが可能になる。現在のDMA設定環境を維持するために、'DMA_Source'のLSBを使用する。

【0103】図18を参照して、'DMA_Source'の最後の2ビット150は、DWord単位でのアドレッシングが行なわれているために使用されていない。この2つのビット152および154のうちLSB154を使用する。このLSBがセットされていた場合、PCI-PCIブリッジは、その後に続く'DMA_Source'アドレスが連続したものであると想定し、内部のソースカウンタを単にインクリメントし続ける。このように連続したアドレスにアクセスするため新たなアドレスフェイズは必要でなく、したがってPCI-PCIブリッジは'DMA_Control'フェイズの後にはサイクルを終了させることなくデータを要求し続けることになる。

【0104】[SSO (Simultaneous Switching Output

ターゲットとして	コンフィギュレーション Read (データフェーズ)
	I/O Read (データフェーズ)
	メモリ Read (データフェーズ)
マスターとして	メモリ Read/Write (アドレスフェーズ)
	メモリ Write (データフェーズ)
	アイドル (If parked on Master)

30

【0108】PCI-PCIブリッジが2次PCIバスのADラインを駆動するのは次の場合である。

【0109】

【表2】

マスターとして	コンフィギュレーション Read (アドレスフェーズ)
	コンフィギュレーション Write (アドレスフェーズおよびデータフェーズ)
	I/O Read (アドレスフェーズ)
	I/O Write (アドレスフェーズおよびデータフェーズ)
	メモリ Read (アドレスフェーズ)
	メモリ Write (アドレスフェーズおよびデータフェーズ)

【0110】2次PCIバスでは、ボード上の「プルアップ」によりADラインが浮遊状態となることは避けられる。したがって2次側では、ADラインが'IDLE'状態にある場合には、2次側のFSMはAD線を駆動する必要はない。

【0111】2次側のADラインを駆動するソースは3つある。その1番目はブリッジデータバスと呼ばれるものであって、コンフィギュレーションとI/Oサイクルに用いられる。またFIFOがディスエーブルされてい

ts) およびデータフロー制御] ところで、プライマリI/O (32ビット) と2次I/O (32ビット) の双方が同時にA_Dラインをスイッチさせた場合、次のような問題が生ずる。仮にこれら64ビットが同時に活性化された場合には、その結果として生ずる電源の電圧変動のためにPCI-PCIブリッジが誤動作する危険性が高い。仮にA_Dバスが浮遊状態であるか、またはトグルしない場合、すなわち同じ値を保持する場合には消費電力はほとんどないので、PCI-PCIブリッジがこれら2つのバスの駆動をこのように効率的に制御することが重要である。

【0105】したがって特別な機能を有するSSOアービタをPCI-PCIブリッジ内に設ける。SSOアービタはプライマリADラインまたは2次ADラインのいずれか一方のみに駆動の許可を与える。これによりPCI-PCIブリッジが電源の電圧変動により誤動作するおそれは少なくなる。

【0106】PCI-PCIブリッジがプライマリPCIBバスのADラインを駆動するのは次の場合である。

【0107】

【表1】

るときにはこれらをバイパスする。2番目はホストからのポスティングに用いられるP2S (Primary-to-Secondary) データバスである。3番目はGEコアからのポスティングに用いられるC2S (Core-to-Secondary) データバスである。

【0112】これらに加えてプライマリADラインを駆動するソースとして独立のものが2つある。その1つはホスト、他の1つはDMAである。ホストは図20に示されるように、プライマリI/O62内のターゲットF

50

SM72のことをいう。DMAに対応するのはプライマリ1/O62内のマスタFSM74である。プライマリFSMと2次FSMすなわちコアとは異なるクロック領域に属するので、2次1/O64に設けられるSSOアービタ(SSOFSM)78は、非同期ハンドシェイキングを行わなければならない。なおこのようにプライマリPCIバスと2次PCIバスとで異なる周波数を用いるのは、同期式に比べる非同期の場合が速度的に有利なためである。また一般的に2次PCIバスの動作するクロック周波数の方がプライマリPCIバスよりも高いからである。

【0113】PCI-PCIブリッジ内での円滑なかつばらnsのとれたデータフローを実現するために、次のような方式を採用した。

【0114】すなわち図20に示されるように、SSOアービタとホストおよびDMAとの間には2つの独立した非同期ハンドシェイクが行なわれる。図20において各非同期ハンドシェイクには同期ポイント162および166ならびに164および168が設定される。なお図20にも示されるように、プライマリ1/O62はプライマリ側のクロック@P_CLKで、2次1/O64のSSOアービタは2次側のクロック@S_CLKでそれぞれ駆動される。

【0115】ハンドシェイクは、各プライマリ状態マシンからのリクエスト信号と、それに対応するSSOアービタからの許可信号とからなっている。ここで生ずるのは主としてプライマリターゲットへの書込アクセスであって、読出アクセスは多くの場合タイミングに厳しくないため、2次PCIバスへのデータソース、主として2つのポスティングバッファの間で円滑かつ高速にローテーションができるよう最適化設計されている。

【0116】したがって、SSOアービタは同期のリクエストと許可とによるペナルティなしにこれらソースの間をローテートする必要がある。たとえばコアとホストとが互いに競合する場合であっても、2次PCIバスからデータを非常に効率的に取出すことができる。

【0117】図21を参照して、SSOアービタ(SSOFSM)78は3つのリクエスト入力と3つのリクエスト出力とを有する。SSOFSMは、自己がどの状態にいるのかに従って、リクエストのうちの1つを2次FSM76に伝えるか、またはプライマリFSMの1つに対してADラインを駆動する許可が与えられている場合には、アービタはADラインがともに同時に駆動されるのを防ぐために3つのリクエスト入力をすべてマスクするか、のいずれかを行なう。このために2次FSM76からSSOFSM78に対して信号'ROTATE'を与える。この信号'ROTATE'と各リクエスト信号とのアンド(182、183、184)をとることにより上述したように2つのADラインが同時に駆動されるのを防ぐ。

【0118】図21に示される信号'ROTATE'は2次FSMが生成するものであるが、この信号によってSSOFSMは、次の状態への遷移をすべきか否かをチェックすることができる。このローテーションは、2次PCIバス上でのデータ転送の実行サイクルごとにリクエストされる。したがってリトライの場合にはローテーションが強制的に行なわれることはない。

【0119】図22にSSOFSMの状態遷移図を示す。この図において各円(HOST, DMA, P2S, C2S, BRIDGE, IDLE)は各状態を示し、状態間の矢印は或るイベントが生じたときの状態遷移を示す。矢印に括弧付きで付された符号(b1, p2など)は、遷移の発生点と、その遷移の優先順位とを示す。遷移の発生点は図22に示すとおり符号を用いて示される。また優先順位とは、同時に複数のリクエストが発生した場合に、ADラインを駆動するための許可が「公平に」行なわれるように定められたものである。優先順位を示す数字は、小さいほど優先順位が高いことを示す。このように適切な優先順位を各遷移に割当てることによりどのソースに対してもADラインの許可が公平に割当てられ、データ転送が円滑に行なわれる。

【0120】図22に示されるSSOFSMによる状態遷移の例を以下に示す。

(1) SSOFSMが'IDLE'にあり、ソース'P2S'および'C2S'が同時に検知された場合には、状態'P2S'への遷移'i1'と状態'C2S'への遷移'i2'とが候補となる。これらのうち前者の方が優先順位が高いのでこの結果状態'P2S'への遷移'i1'が生ずることになる。

【0121】(2) SSOFSMが状態'BRIDGE'にあり、かつSSOFSMがローテートされ、かつソース'P2S'および'HOST'からのリクエストがちょうどこのときに同時に発生したものとすると、この場合には状態'HOST'への遷移'b1'と状態'P2S'への遷移'b3'とが発生し得るが、前者の方が優先順位が高いので状態'HOST'への遷移'b1'が生ずることになる。

【0122】図22において、優先順位の最も高い遷移が外周を時計回りに回っている点に注意されたい。このようにすることで、ローテーションが「公平に」行なわれる。もちろん、図22に示される例は単なる一例であって、これ以外の状態遷移であっても同様の効果を奏するものであればよい。

【0123】[フロー制御] トランザクションが完了するごとにローテーションをした場合には効率が落ちる場合もあるので、ポスティングバッファ'P2S'および'C2S'については、他のソースからのリクエストが発生している場合であってもいくつかのトランザクションにわたってADライン使用の許可を保持することができるようになる。この様子を図23に示す。

【0124】このために、図23に示されるように信号' ROTATE' を活性化するタイミングを適切に制御する。この信号はデータ伝送の正常な終了を示すものであり、リトライの場合には活性化されない。図23に示される例では、リトライが生じた時点ではこの信号は活性化されず、サイクル終了時に活性化されている。

【0125】たとえばメモリ書込について長いバーストをサポートするシステムでは、2次PCIバスに対してデータを転送する際にはポスティングが効率的であろう。

【0126】[P2Sバースト制御] ポスティングバッファによってトリガされたサイクルは、メモリバンクのサイズに従って2次PCIバス上では分割される。すなわち仮に16DWordバーストがポストされた場合には、2次PCIバスではそれは2つの8DWordサイクルに分割される。デフォルトではSSOFSMは最初の8DWordの書込が行なわれかつ他のリクエストが生じている場合にローテートすることになる。また、小さなブロックのデータしかポストされなかったときには、読出されたDWordの数にかかわらずSSOFSMはローテートするであろう。たとえば2つの1DWordサイクルがポストされた場合には、デフォルトではSSOFSMは最初のDWordが書込まれかつ他のリクエストが生じた場合にローテートすることになる。

【0127】しかし' P2S' バッファが、他の状態にローテートするまでにいくつかのサイクルを実行できるようにした方がよい場合もある。そこでこの実施の形態のPCI-PCIバスではMMPCIレジスタ' P2S_Burst' を設け、このレジスタによってこれを制御するようにしている。このレジスタの値は、' P2S' バッファがSSOアービタのローテートするまでに実行可能なサイクル数を決定する。

【0128】たとえば図24(A)を参照して、' P2S' のポスティングデータ、たとえば16DWordは、デフォルトでは2つの8DWordとして分割される。そのためこの2つの8DWordの間には時間的なギャップが生ずる。もし' P2S' のポストするデータに対しレイテンシを低くしたい場合には、レジスタ' P2S_Burst' を用いてこのポスティングバッファに対し高い優先順位を割当てることができる。すなわち図24(B)に示すようにレジスタ' P2S_Burst' を「2」に設定すれば、' P2S' のポスティングバッファは、ADラインを駆動するための許可を戻すことなく、2次PCIバス上に最後のデータまで(2個)一度に書込むことができる。これにより' P2S' のポストするデータに対するレイテンシが低くなる。

【0129】[C2Sバースト制御] 同様の制御を' C2S' ポスティングバッファに対しても適用することができる。' P2S' の場合にはメモリバンクとして2つある場合を想定したが、本実施の形態のPCI-PCI

ブリッジでは' C2S' ポスティングバッファとして4つのメモリバンクを有するものを想定している。その点で' C2S' ポスティングバッファは' P2S' と異なるが、さらに「GEコア」というソースを持つ点でも' P2S' ポスティングバッファと相違している。' C2S' ポスティングバッファは2次PCIデバイスの専用データを含んでおり、データブロックがメモリバンクのいくつかに分散している場合でもまとめて転送できるようにした方が好ましい。こうした制御はこの実施の形態のPCI-PCIブリッジではMMPCIレジスタ' C2S_Burst' を用いて行なうことができる。すなわちこのレジスタ' C2S_Burst' は、SSOアービタがローテートするまでに' C2S' バッファが実行することができるサイクル数を決定する。

【0130】たとえば図25(A)を参照して、レジスタ' C2S_Burst' が1に設定されていれば' C2S' ポスティングバッファは1サイクルごとにADラインの使用許可を戻さなければならない。レジスタ' C2S_Burst' が3に設定されている場合には、図25(B)に示されるように3サイクルまで連続して実行することができる。このように' C2S' ポスティングバッファが連続して実行することができるサイクル数を制御できるようにしたことにより、いくつかのメモリブロックからなるデータをGEコアから2次PCIデバイスに一括して転送することができる。

【0131】[DMAバースト制御] 既に述べたようにSSOアービタは2次PCIバス側のクロック領域に設けられている。つまり2次PCIバス側のクロック信号で動作している。そのためSSOアービタはDMAFSMとの間で完全な非同期ハンドシェイクをする必要がある。その結果SSOFSMにおいて状態が' DMA' 状態に変化したり、' DMA' 状態から変化したりする場合にはいつも、同期ペナルティを考慮しておく必要がある。

【0132】DMAバーストが非常に短い場合、すなわちホストブリッジが小さなバッファしか持たない場合には、同期ペナルティにより全体的なスループットの遅延が増加する。そのため' P2S' 状態にローテートする前に、DMAFSMがいくつかのバーストを行なうようにした方が効率的な場合がある。これをこの実施の形態のPCI-PCIバスではMMPCIレジスタ' DMA_Burst' により制御することにした。レジスタ' DMA_Burst' は、DMAFSMが、リクエストをSSOアービタに譲るまでに実行できるサイクル数を決定する。

【0133】図26(A)に、DMAトランザクションが' HOST' によって分断されている状態を示す。この場合にはDMAFSMは、もう一度サイクルを発行しなければならない。これによりDMAトランザクションの全体的なレイテンシは同期ペナルティ(図26にお

10

20

30

40

50

いて斜線で示される)により2倍となる('DMA_Burst'が1に設定されている場合)。しかし図26(B)に示されるように、'DMA_Burst'が2に設定されている場合には同期ペナルティは最小化され全体として処理時間は図26に t_{diff} で示される時間だけ短縮できる。

【0134】[ポスティングFIFOのパラメータ化]
今まで述べてきた'P2S'ポスティングバッファ、'C2S'ポスティングバッファなどはFIFOにより構成される。ポスティングFIFOの設計は、結局のところ利用可能な面積と、必要とされるスループットおよび許される最大のレイテンシの間の妥協の産物である。こうした条件によってポスティングトランザクションのためのFIFOの粒度が決定される。環境により課される制約が種々にわたるため、FIFOを実装する際に多くのアーキテクチャが可能である方が有用である。そこで本実施の形態のPCI-PCIブリッジでは、FIFOの設計をパラメータ化する手法を採用する。

【0135】図27に、パラメータ化されたポスティングFIFOのブロック図を示す。図27を参照してこのポスティングFIFOは、いずれもFIFOの動作を制御するための、小規模な読出ポートFSMおよび書込ポートFSMと、これら2つのFSMの間に挟まれたいくつかの「スライス」からなる。このようにいくつか並べることによりポスティングバッファを形成する。

【0136】図28を参照して、各「スライス」はデータを格納するためのnDWordのサイズを持つメモリバンクと、対応する(開始)アドレスを格納するための32ビットレジスタと、「フル」フラグおよび「エンベティ」フラグと、これら2つのフラグについての完全非同期ハンドシェイクとを含んでいる。

【0137】このように2つのFSMの間にメモリバンクのスライスを挟む構造でFIFOを形成したことにより、スライスの数というパラメータを所望のものに合わせて選択することにより容易にポスティングFIFOの設計を行なうことができる。たとえばハードウェア記述言語を用いて設計を行なう場合、スライス数に相当するパラメータを単に仕様に依じて変更するだけで足りる。そのため仕様の変更や仕様の改訂に柔軟に対応することができるとともに、複数のポスティングバッファの検証も非常に簡略に行なえるという特徴がある。

【0138】なお、本願発明を一実施の形態に基づいて説明してきたが、これは一つの例に過ぎず、本願発明がこの実施の形態の装置に限定されるわけではないことはもちろんである。

【0139】

【発明の効果】以上のようにこの発明によれば、タイプ「01」ヘッダを用いないPCI-PCIブリッジにより、2次PCIバスを持つ半導体集積回路装置をプライマリPCIバスに接続し、かつ個別のPCIエージェン

トを独立に制御できる。。

【0140】また、VGAデバイスとPCIエージェン

トとが接続された2次PCIバスと、プライマリPCIバスとの間を接続可能なPCI-PCIブリッジを提供することができる。

【0141】さらにこの発明によれば、起動時のエラーメッセージの表示を、既存のVGAデバイスを用いて容易に行なうことができ、また一旦デバイスドライバが起動すれば、PCIデバイスを用いた処理を行なうことができる。

【0142】さらにこの発明によると、PCI-PCIブリッジにより効率的なDMA転送が可能になる。また主記憶と、2次PCIバス上のPCIエージェン

トまたは特定の演算を行なう演算回路との間でのデータ転送を高速で行なうことができる。

【0143】さらにこの発明によると、FIFOの大きさをパラメータに還元し、したがって容易に設計変更を行なうことが可能なので、FIFOを有するPCI-PCIブリッジの設計を容易にすることができる。

【0144】請求項12に記載の発明にかかるPCI-PCIブリッジは、請求項11に記載のFIFOを用いた、請求項5~9のいずれかに記載のPCI-PCIブリッジである。

【0145】さらにこの発明によるPCI-PCIブリッジでは、データ転送を高速に、効率よく行なえる上、データの転送の際に用いられる先入れ先出しメモリの大きさを容易に変更できるため、仕様変更や、突然の設計変更があってもPCI-PCIブリッジの設計を簡単に修正することができ、装置の開発期間を短くすることができる。

【図面の簡単な説明】

【図1】 本発明の一実施の形態に係るPCI-PCIブリッジを含んだシステムの全体構成を示すブロック図である。

【図2】 本願発明の一実施の形態に係るPCI-PCIブリッジのブロック図である。

【図3】 タイプ「00」ヘッダのビット構成を示す図である。

【図4】 ファンクション番号と、対応するデバイスとの関係を表形式で示す図である。

【図5】 VGAデバイスが含まれるか否かを示すスラップ信号の値の意味を表形式で示す図である。

【図6】 シャドーされたレジスタの読出を示す図である。

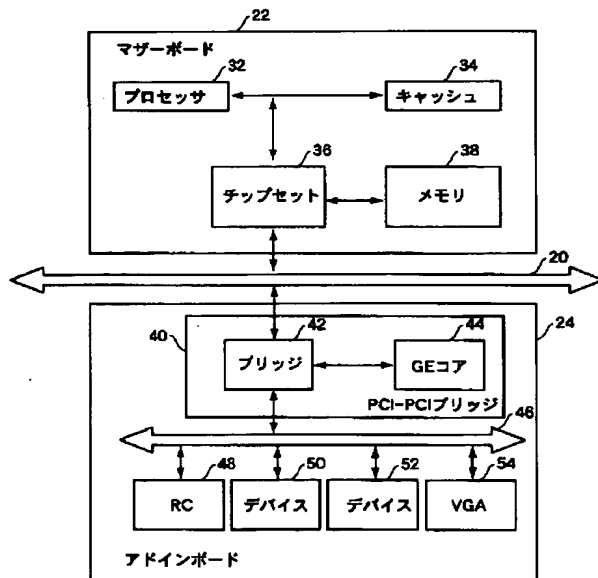
【図7】 シャドーされたレジスタに対する書込を示す図である。

【図8】 レジスタへの書込の際のマスキングを示す図である。

【図9】 レジスタへの書込の際のマスキングを示す図である。

- 【図10】 レジスタへの書込の際のマスキングを示す図である。
- 【図11】 一般的なメモリマップを模式的に示す図である。
- 【図12】 VGAメモリマップを模式的に示す図である。
- 【図13】 ブリッジとVGAデバイスとの間のIDSELの接続関係を示す図である。
- 【図14】 DMA転送をセットアップする際のレジスタを表形式で示す図である。
- 【図15】 本発明に係るDMA転送のオンザフライコンフィギュレーション方式を示す図である。
- 【図16】 本発明に係るDMA転送のオンザフライコンフィギュレーション方式を示す図である。
- 【図17】 本発明に係るDMA転送のオンザフライコンフィギュレーション方式を示す図である。
- 【図18】 本発明に係るDMA転送のオンザフライコンフィギュレーション方式を示す図である。
- 【図19】 リンクデータの'DMA_Source'の下位2ビットを示す図である。
- 【図20】 プライマリI/O内のFSMとSSOFSMとの非同期ハンドシェイキングを示す図である。

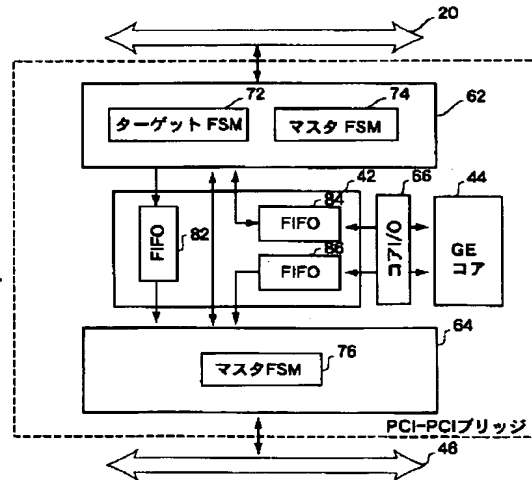
【図1】



20:プライマリPCIバス
46:2次PCIバス

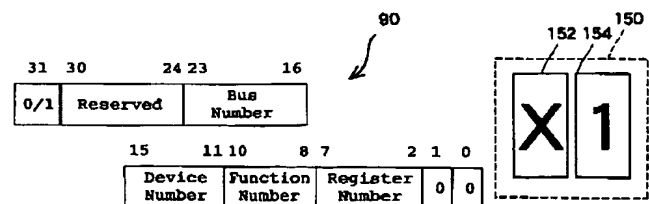
- 【図21】 SSOFSMと2次側FSMとの間の信号の入出力関係を示す図である。
- 【図22】 SSOFSMの状態遷移図である。
- 【図23】 'P2S'ポスティングを複数サイクル連続して許可するようにした場合の波形図である。
- 【図24】 'P2S_Burst'レジスタの値とデータ転送のタイミングとの関係を示す図である。
- 【図25】 'C2S_Burst'レジスタの値とデータ転送のタイミングとの関係を示す図である。
- 10 【図26】 'DMA_Burst'レジスタの値とデータ転送のタイミングとの関係を示す図である。
- 【図27】 本発明に係るポスティングFIFOバッファの構成を模式的に示す図である。
- 【図28】 ポスティングFIFO内の位置スライスの構成を示す図である。
- 【符号の説明】
- 20 プライマリPCIバス、24 アドインボード、40 PCI-PCIブリッジ、42 ブリッジ部、44 GEコア、46 2次PCIバス、54 VGAデバイス、62 プライマリI/O、64 2次I/O、72 プライマリターゲットFSM、74 プライマリマスタFSM、76 2次マスタFSM。

【図2】



【図3】

【図19】



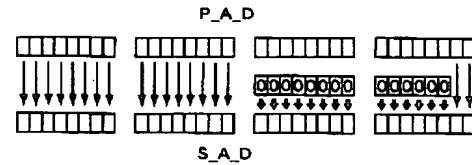
【図 4】

000 b	PCI-PCI Bridge	-
001 b	S_A_D [28]	Device 1
010 b	S_A_D [29]	Device 2
011 b	S_A_D [30]	VGA Device
100 b	S_A_D [31]	Device 3
101,110,111 b	PCI-PCI Bridge	-

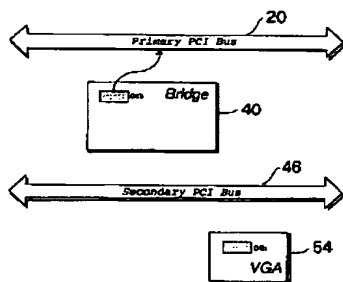
【図 5】

0 b	off (Non-VGA Device)
1 b	on (VGA Device)

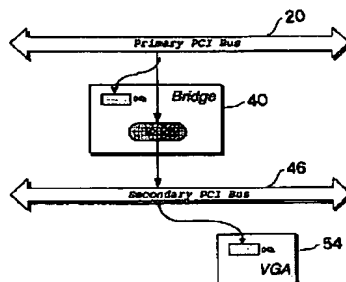
【図 8】



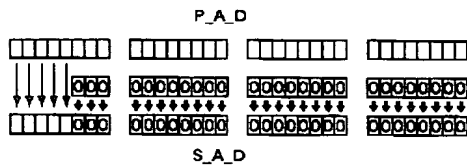
【図 6】



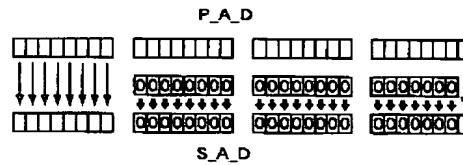
【図 7】



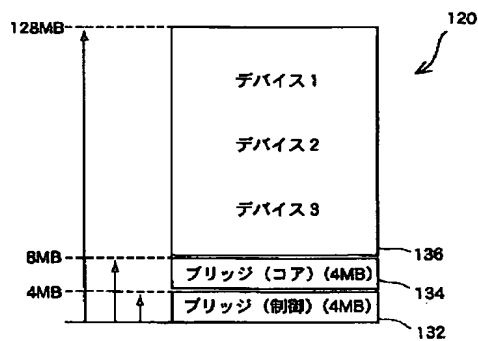
【図 9】



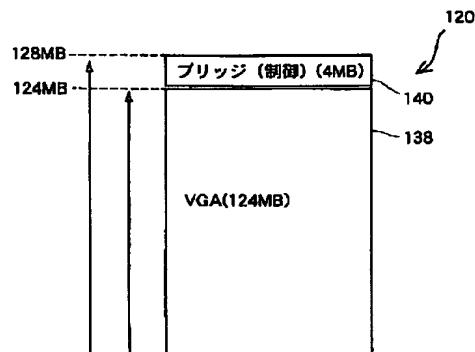
【図 10】



【図 11】



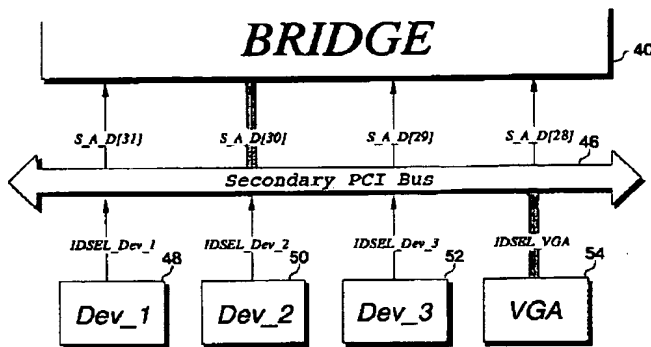
【図 12】



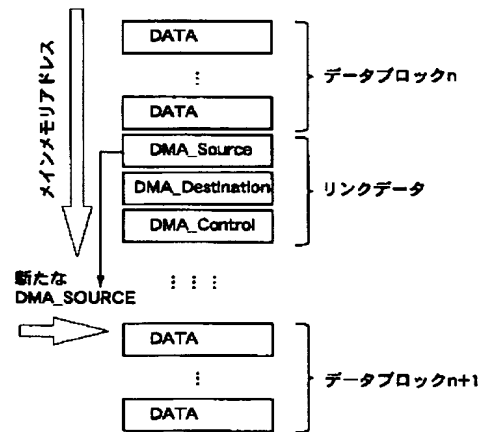
【図 14】

1	DMA_Source	064b
2	DMA_Destination	068b
3	DMA_Control	06Ch

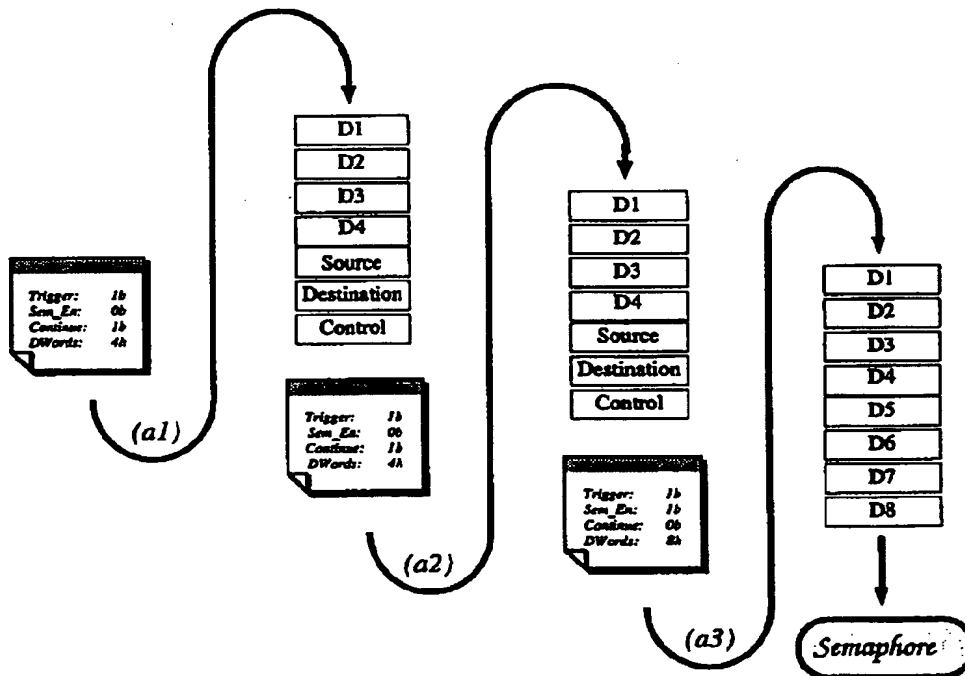
【図13】



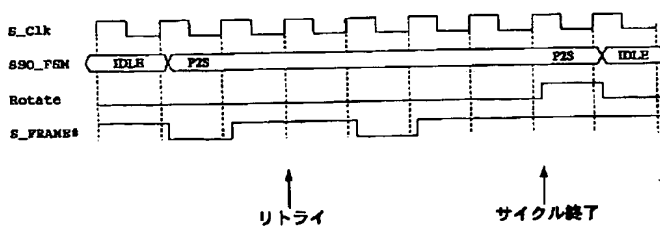
【図15】



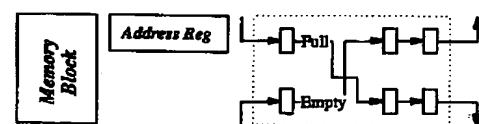
【図16】



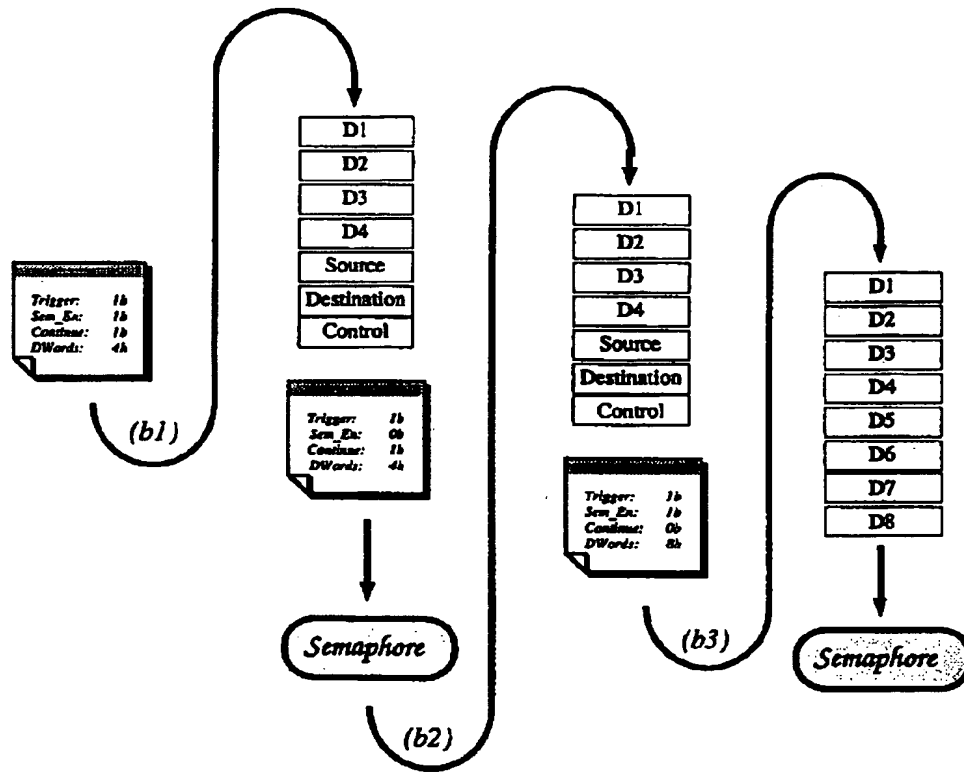
【図23】



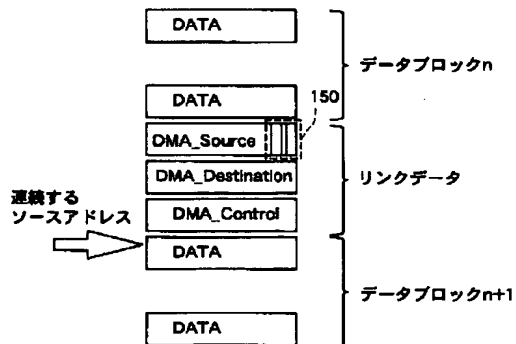
【図28】



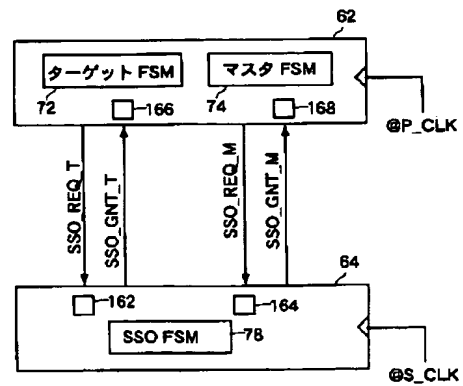
【図17】



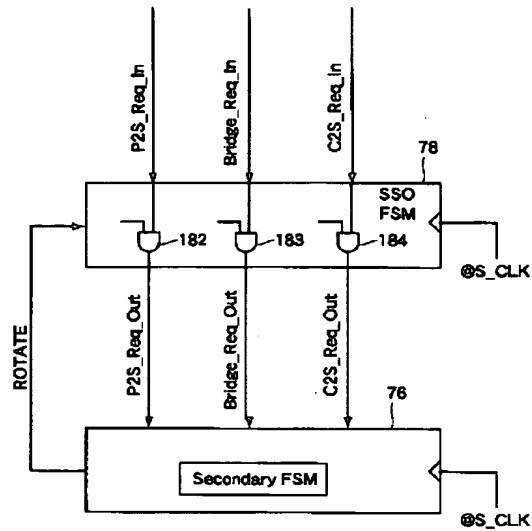
【図18】



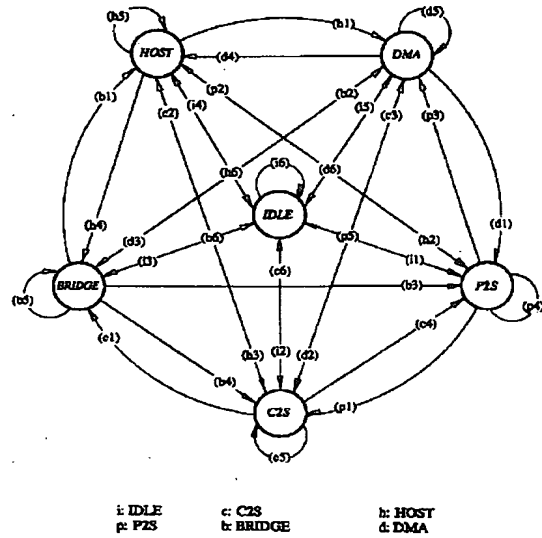
【図20】



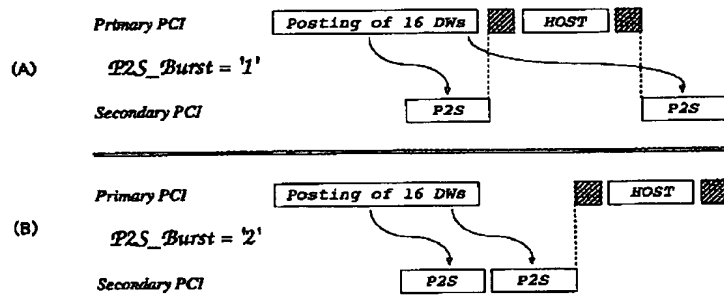
【図 2 1】



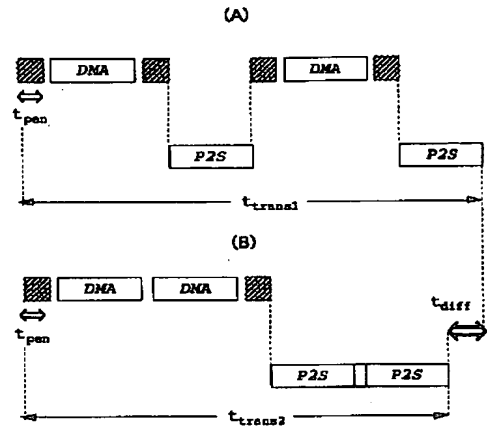
【図 2 2】



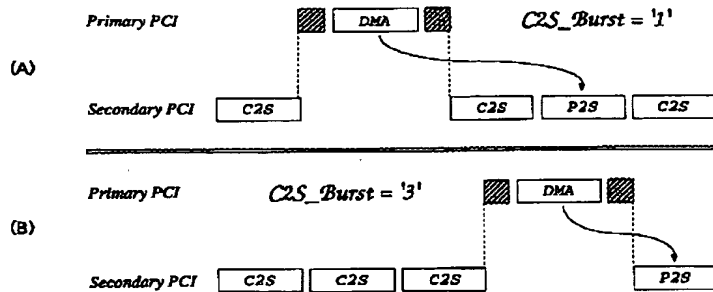
【図 2 4】



【図 2 6】



【図 2 5】



【図 2 7】

